**Foundation**

SQL

Module 11

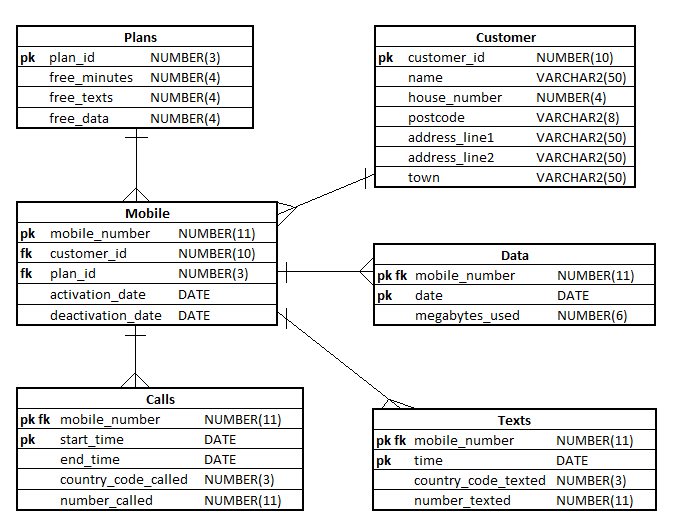
Filtering Aggregates– Query reading

© FDM Group Ltd 2017. All Rights Reserved.

Any unauthorised reproduction or distribution in part  
or in whole will constitute an infringement of copyright.

**Document change and version control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Reason for Change** | **Author** | **Date of Change** | **New Version No** |
| Creation of document | Nick Lawton | 26/06/2017 | v1.0 |
| A mistake in question 4 rectified: the query groups records by customers’ name, thus if there are more customers with the same name it would calculate the number of mobile phones for all of them together, rather than separately. The problem is fixed by including the customer\_id to both SELECT and GROUP BY clauses. | Nikola Ignjatovic | 21/07/2021 | v1.1 |
| Question 6 added, to include an example of a query that returns a calculated value that involves an aggregate function and that includes a condition involving an aggregate function | Nikola Ignjatovic | 30/07/2021 | v1.2 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

****

**Exercise 1**

## Describe in simple language what each of the following queries will return:

|  |
| --- |
| **Questions** |
| 1. SELECT mobile\_number   FROM data  GROUP BY mobile\_number  HAVING SUM(megabytes\_used) > 1000000;  Mobile numbers that have used over 1000000 megabytes |
| 1. SELECT mobile\_number   FROM texts  GROUP BY mobile\_number  HAVING COUNT(\*) > 100000;  Mobiles that have sent over 100000 texts |
| 1. SELECT mobile\_number   FROM texts  WHERE country\_code\_texted != 44 **(Country code 44 is the UK)**  GROUP BY mobile\_number  HAVING COUNT(\*) > 1000;  Mobile numbers that have sent 1000 texts to numbers not in the uk. |
| 1. SELECT c.customer\_id,   c.name  FROM customer c  INNER JOIN  mobile m  ON c.customer\_id = m.customer\_id  GROUP BY c.customer\_id,  c.name  HAVING COUNT(m.mobile\_number) > 3;  Show customer name and id that have greater than 3 mobile numbers |
| 1. SELECT mobile\_number   FROM data  GROUP BY mobile\_number  HAVING AVG(megabytes\_used) >  (  SELECT AVG(megabytes\_used)  FROM data  );  The numbers that have a greater than average megabite usage |
| **6**. SELECT m.mobile\_number,  SUM(NVL(d.megabytes\_used, 0)) - NVL(p.free\_data, 0)  FROM plans p  INNER JOIN  mobile m  ON m.plan\_id = p.plan\_id  INNER JOIN  data d  ON d.mobile\_number = m.mobile\_number  WHERE TO\_CHAR(d.date, 'YYMM') = TO\_CHAR(SYSDATE, 'YYMM')  GROUP BY m.mobile\_number  HAVING SUM(NVL(d.megabytes\_used, 0)) > NVL(p.free\_data, 0);  Mobile number that have gone over their monthly usage of free data |